CLAIMS

What is claimed is:

1. A method of obscuring cryptographic computations comprising: performing modular exponentiation in a cryptographic computation such that memory accesses are independent of the numerical value of the exponent.

2. The method of claim 1, wherein performing modular exponentiation comprises replacing a conditional multiplication operation with an unconditional multiplication operation.

3. The method of claim 2, wherein the unconditional multiplication operation uses an obscuring factor.

4. The method of claim 3, further comprising wherein for each bit in the exponent, determining the obscuring factor by multiplying a quantity by a selected bit of the exponent plus one, the quantity comprising a message minus one.

5. The method of claim 1, wherein the exponent comprises at least one of a signature exponent and a decryption exponent in a RSA cryptographic system, and the cryptographic computation is at least one of signature and decryption.

6. The method of claim 5, wherein the cryptographic computation comprises $\mathbf{c}^{d}$ mod **n**, wherein **c** comprises a ciphertext message, **d** comprises the decryption exponent, and **n** comprises a modulus that is a product of two prime numbers.

7. The method of claim 1, wherein the modular exponentiation is performed as part of a Diffie-Hellman key exchange process.

8. The method of claim 1, wherein the modular exponentiation is performed as part of a Digital Signature Algorithm (DSA) process.

9. The method of claim 1, further comprising applying a window method as part of performing the modular exponentiation and retrieving pre-computed powers from one to 2^v of a message from a memory, where v is the size of a window into the exponent's bits.

10. A method of obscuring cryptographic computations by performing modular exponentiation of an exponent in a cryptographic computation such that memory accesses are independent of the exponent bit pattern comprising:

setting an intermediate value to a message; and

for each bit i in the exponent, setting the intermediate value to the intermediate value multiplied by the intermediate value mod a modulus, wherein the modulus comprises a product of two prime numbers, determining a current obscuring factor using the i'th bit of the exponent, and setting the intermediate value to the intermediate value multiplied by the current obscuring factor mod the modulus.

11. The method of claim 10, wherein determining the current obscuring factor comprises determining the current obscuring factor by multiplying a quantity by a selected bit of the exponent plus one, the quantity comprising the message minus one.

12. The method of claim 10, wherein the exponent comprises at least one of a signature exponent and a decryption exponent in a RSA cryptographic system, and the cryptographic computation is at least one of signature and decryption.

13. The method of claim 10, further comprising applying a window method as part of performing the modular exponentiation and retrieving pre-computed powers from one to 2^v of the message from a memory, where v is the size of a window into the exponent's bits.

14. A method of obscuring cryptographic computations by performing modular exponentiation of an exponent in a cryptographic computation such that memory accesses are independent of the exponent bit pattern comprising:

picking a random number between one and a modulus minus one, the modulus comprising a product of two prime numbers;

determining an intermediate value based at least in part on the random number and a message;

determining a first obscuring factor and a second obscuring factor using the message and the inverse of the random number;

for each bit i in the exponent, setting the intermediate value to the intermediate value multiplied by the intermediate value mod the modulus, determining a third obscuring factor using the i'th bit of the exponent and the first and second obscuring factors, and setting the intermediate value to the intermediate value multiplied by the third obscuring factor mod the modulus; and

setting a new message to the intermediate value multiplied by the second obscuring factor mod the modulus.

15. The method of claim 14, wherein determining the intermediate value comprises setting the intermediate value to the message multiplied by the random number mod the modulus.(it's line 7, there is no line 14 in Table III)

16. The method of claim 14, wherein determining the first obscuring factor comprises setting the first obscuring factor to the message multiplied by the inverse of the random number mod the modulus.

17. The method of claim 14, wherein determining the third obscuring factor comprises setting the third obscuring factor to

(w AND $\mathbf{d}_i$) OR (**s** AND (NOT $\mathbf{d}_i$)), wherein w is the first obscuring factor, $\mathbf{d}_i$ is the i'th bit of the exponent, and **s** is the second obscuring factor.

18. The method of claim 14, wherein the exponent comprises at least one of a signature exponent and a decryption exponent in a RSA cryptographic system, and the cryptographic computation is at least one of signature and decryption.

19. The method of claim 18, wherein the cryptographic computation comprises $\mathbf{c}^{\mathbf{d}}$ mod **n**, wherein **c** comprises a ciphertext message, **d** comprises the decryption exponent, and **n** comprises the modulus.

20. The method of claim 14, wherein the modular exponentiation is performed as part of a Diffie-Hellman key exchange process.

21. The method of claim 14, wherein the modular exponentiation is performed as part of a Digital Signature Algorithm (DSA) process.

22. The method of claim 14, further comprising applying a window method as part of performing the modular exponentiation and retrieving pre-computed powers from one to 2^v of the message from a memory, where v is the size of a window into the exponent's bits.

23. The method of claim 22, further comprising multiplying each of the powers of the message by $\mathbf{s}^{(2^\wedge v - 1)}$ mod the modulus, where s = t^{-1} MOD nand **t** is the random number.

24. An article comprising: a storage medium having a plurality of machine readable instructions, wherein when the instructions are executed by a

processor, the instructions provide for obscuring cryptographic computations by performing modular exponentiation of an exponent in a cryptographic computation such that memory accesses are independent of the numerical value of the exponent.

25. The article of claim 24, wherein performing modular exponentiation comprises replacing a conditional multiplication operation with an unconditional multiplication operation.

26. The article of claim 24, further comprising instructions for applying a window method as part of performing the modular exponentiation and retrieving pre-computed powers from one to 2^v of a message from a memory, where v is the size of a window into the exponent's bits.

27. An article comprising: a storage medium having a plurality of machine readable instructions, wherein when the instructions are executed by a processor, the instructions provide for obscuring cryptographic computations by performing modular exponentiation of an exponent in a cryptographic computation such that memory accesses are independent of the exponent bit pattern, the instructions causing

setting an intermediate value to a message; and

for each bit i in the exponent, setting the intermediate value to the intermediate value multiplied by the intermediate value mod a modulus, wherein the modulus comprises a product of two prime numbers, determining a current obscuring factor using the i'th bit of the exponent, and setting the intermediate value to the intermediate value multiplied by the current obscuring factor mod the modulus.

28. The article of claim 27, wherein determining the current obscuring factor comprises determining the current obscuring factor as multiplying a

quantity by a selected bit of the exponent plus one, the quantity comprising the message minus one.

29. The article of claim 27, wherein the exponent comprises at least one of a signature exponent and a decryption exponent in a RSA cryptographic system, and the cryptographic computation is at least one of signature and decryption.

30. The article of claim 27, further comprising instructions for applying a window method as part of performing the modular exponentiation and retrieving pre-computed powers from one to 2^v of a message from a memory, where v is the size of a window into the exponent's bits.

31. An article comprising: a storage medium having a plurality of machine readable instructions, wherein when the instructions are executed by a processor, the instructions provide for obscuring cryptographic computations by performing modular exponentiation of an exponent in a cryptographic computation such that memory accesses are independent of the exponent bit pattern, the instructions causing picking a random number between one and a modulus minus one, the modulus comprising a product of two prime numbers;

determining an intermediate value based at least in part on the random number and a message;

determining a first obscuring factor and a second obscuring factor using the message and the inverse of the random number;

for each bit i in the exponent, setting the intermediate value to the intermediate value multiplied by the intermediate value mod the modulus, determining a third obscuring factor using the i'th bit of the exponent, and the first and second obscuring factors, and setting the intermediate value to the intermediate value multiplied by the third obscuring factor mod the modulus; and

setting a new message to the intermediate value multiplied by the second obscuring factor mod the modulus.

32. The article of claim 31, further comprising instructions for applying a window method as part of performing the modular exponentiation and retrieving pre-computed powers from one to 2^v of the message from a memory, where v is the size of a window into the exponent's bits.